EXHIBIT A

Alexander Winokur/Haifa/IBM@IBMIL
Amnon Ribak/Haifa/IBM@IBMIL
Chris Molloy/Raleigh/IBM@IBMUS

Please fill out the following. All questions are mandatory.

(1) The Problem: What problem is solved by the invention?

In today's business environment, many applications need to use data that is warehoused in diverse data sources and repositories. The data is expressed in different formats and languages, retrieved in different access methods, and through different delivery vehicles. Moreover, new data sources may be added, and existing ones removed or changed frequently.
This problem is more and more common in areas where applications and databases were developed gradually over many years to supply increase demand inorganisations for Information Systems (IS). Thus, it is critical with organisations that are using IS technology for many years.

Examples of such applications include:

- Analyzing System Performance - reports for analyzing the multifaceted aspects of system performance usually use several data sources. There is a data source that includes measures of servers resources such as CPU utilization, percent of memory used, number of users logged in, etc. Another data source measures the response time of probes (tests) done on the servers such as a round-trip e-mail message, access to specific web pages, etc. A third data source contains network performance measures.
  The analyzing application would like to correlate those data sources and allow the examination of how a probe performs at a specific load on the server and a specific load on the network. The data correlation will allow to examine what caused poor performance of the system.
- Personnel data warehouses - Personnel systems exist in organisations for many years. They were developed gradually with each new requirement that was developed. Thus, it is possible to find very old data (residing in indexed sequential files) holding information about people's demographic information, history of ranks or levels, latest salaries etc. Over the years new databases or files that hold information about education were developed by the internal education department of an organisation. Later on, new applications were developed to create databases that hold information about employees locations such as room, telephone numbers, e-mail address etc. The result is that information about employees are distributed in many data sources, each handled by different application and each holds some unique information and some duplicate information.
- Customer Relationship Management - customer data files with information about customers existed since organisations started to use computers to manage their customers data. Initially, only the minimal information was stored in sequential data files, later new databases were created by new applications that started to store additional information such as billing and customers profiles. Other applications (developed within other areas) created databases that contained orders. Within problem management systems, databases containing trouble tickets

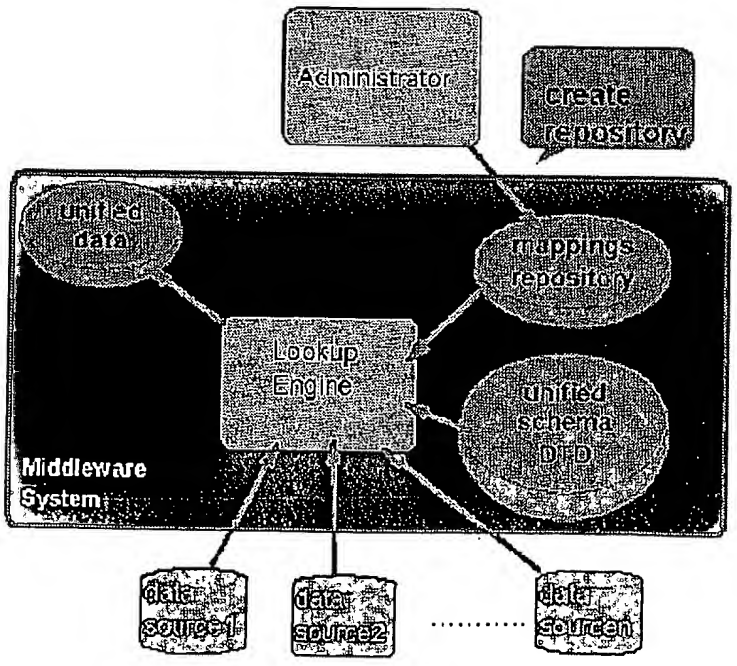
and customer complains were developed. The result is that customer data resides in different data sources.

A method to access and integrate the different data sources is needed. The method should be generic to all domains, and allow the adaptation of changes in the data source easily. The method should provide the application a uniform way to access the data as if there was only one data source. By looking uniformly at the different data sources and seeing how the information is related, the application can give a better breakdown of the entire situation.

(2) Define the Invention: What features, elements, compositions, materials, or process steps, or combinations of these that make up the invention are new?

We present a novel method and a middleware system for integrating diverse data sources using eXtensible Markup Language (XML). The system consists of an Administrator application and a Lookup Engine. A unified schema represented in a Document Type Definition (DTD), and a repository of mappings are created to map from the data sources to the unified schema. Then, the Lookup Engine uses the repository of mappings to extract the relevant data from the data sources and create the unified data. The unified data is represented in XML and complies to the unified schema DTD. All the complexity of accessing the data sources, retrieving the data, and correlating it is done by the Lookup Engine, and is transparent to the application which just uses the output of the Lookup Engine - the unified data.
In this invention, we assume that only reads operations are done on the data sources. Although, it can be extended to include distributed transactions, this is out of the scope of this invention.

The following figure sketches the proposed system:



Basics of operation:

1. Create a **unified schema DTD** for the selected domain. The unified schema is represented in a DTD but may be in the future represented in an XML-Schema. All the relations of the domain data is defined by this DTD. Over time, there will exist DTDs (XML-schemas) for all domains. Thus, one would need just to select its appropriate DTD. A performanceML DTD for the system performance domain and a CPEX DTD for the customer relationship management domain are currently under development.
2. The **Administrator** defines how to access each data source. For example, if the data source is a database, it will define the host name and port, the database name, the user and password. If the data source is a file, it will define the file path and file structure.
3. The Administrator creates **mappings** from the data sources to the unified schema DTD. A mapping is a triplet of the form *(source, target, conversion-function).*
   The source is a field or a set of fields in the data source such as *Day.cpu_utilization* for the cpu_utilization in the Day table.
   The target is an element or an attribute in the unified schema DTD such as the following attribute in PerformanceML DTD -
   PERFORMANCE.Server.Server_Performance_Info.CPU_utilization.
   The conversion function is a function to be applied on the data source to create the value for the target, such as *floatToPercentage.*
   Note that the source doesn't have to be a field in a database. It can be a field in any data source as long as the information to uniquely identify it is given in the mapping.
4. The **Lookup Engine** creates unified data according to the mappings and the unified schema DTD. For each mapping, it gets the values from the data source, activates the conversion function, and creates the XML element or attribute. The output of the Lookup Engine is the unified data represented in XML which complies to the unified schema DTD.
5. Next, an XML query engine can be applied on the unified data. XQL and XML-QL are two XML query languages that have query engines under development.

Adding/ Removing/ Changing a data source:

Since the DTD for a domain is developed and evaluated by the whole community, it is (will become eventually) a mature domain DTD. Consequently, the unified schema DTD is not expected to change when the data sources change. The Administrator needs to add, remove, or update mappings when a new data source is introduced, removed or changed. However, the applications themselves don't have to change. Moreover, we can assume that over time as the domain DTD becomes a standard, the data sources will use a similar language to that defined by the domain DTD. This will ease the work of the Administrator when constructing new mappings.

Performance issues:

There are some concerns regarding the performance of creating the unified data in XML, and querying it using an XML query engine. We would like to point out that a lot of work is done in the industry today on XML in general, and XML performance issues in particular. Thus, it is expected to achieve good performance on creating large XML data and querying it.

Problems addressed by this invention: claims

There are several features which may be considered as independent claims or genus with respect to each other:

1. XML Unified Schema - for each domain, it is possible to define a common unified schema that will establish a standard for domain specific record and will present a union of all possible information that may be acquired on items in the domain. XML schema can be hierarchical with optional parts.

2. Generic Mapping tools - the ability to create mappings between multiple data sources and common XML schema combined with conversion functions, will enable developers the flexibility to add data sources of any type without any restrictions. This is a very powerful solution that can be enhanced with "on the fly" mappers. There are three main characteristics unique in this invention: the generic of the mapping tool, the ability to add/remove mappings dynamically, and the ability to provide customized conversion functions.

3. Mapper Lookup - since applications and specifically Web based applications, can be created dynamically, it is important to be able to provide a dynamic lookup engine that can find mappings on the fly and can always use new mappers that are added dynamically. Unlike many systems, that may provide some static mappings or create joined data bases that must be created in advance, this invention provides a way to integrate data dynamically without a need to join data in advance in databases.

(3) Alternatives: What are the alternate solutions practiced now or possible in the future?

Traditional solutions to integrating multiple data sources is to create a new data warehouse and copy the data from the original diverse data sources to the warehouse. This solution is not flexible to dynamic changes in the data sources. IBM DataJoiner is a product that employs this solution. Also, invention US5884310 is based on this type of solution.

Invention US5345586 solves the problem using a global data directory which maps the location of data, specific data entity attributes and data source parameters. Our method is different in the use of a domain DTD and conversion functions that are not used in any existing solution or invention.

There is no prior work on data integration using XML.

(4) Advantages: What technical advantages does the invention have over existing or possible alternate solutions?

Here are some of the advantages:

- The proposed method and system present a middleware to unify diverse data sources such that the data unification is transparent to the application. The application feels and acts as it has just one data source.

- In the proposed method and system, it is easier to add, remove, or change data sources. Only the Administrator need to change the mappings. The applications doesn't have to change.
- In the proposed method and system, the data source can be of any type, and not just a relational database.
- In the proposed method and system, the unified data is represented in XML, and thus more easily can be exchanged in a web environment. XML is rapidly becoming accepted as the standard for data interchange, across the web and between applications.
- In the proposed method and system, the conversion functions used in the mappings are general and can be reused from one system to another.

(5) Operability: How has the invention been demonstrated to work either physically or logically?

The basics of operation in (2) describe how to implement the system and method. The Lookup Engine is the most difficult part and it should be probably built in stages. We should create in the Lookup Engine a general interface and then incrementally add implementations of the interface; each time for a different type of data source.
The Lookup Engine can be either triggered by a scheduler that periodically updates the unified data or by an external event such as a query on the unified data.

There is yet no system available to demonstrate the invention physically. There exists a prototype.

(6) Importance to Others:

What are the commercial benefits of the invention?

- Implement as an IBM data integration product.
- Implement in IBM products that need data integration, and there are a lot of them.
- License to data integration vendors.

How feasible is it to implement the invention commercially?

see (5)

Which industries (and companies) outside of IBM would be likely to need the invention?

Every company which needs to integrate diverse data sources, or a company that develops data integration technology such as ORACLE.

Identify the products for which the invention has application.

Products and services that use diverse data sources, or data integration products.
In (1) we presented two examples of products that use diverse data sources.

How can the invention be detected if used by others?

If the product contains an Administrator application to create mappings from data sources to a DTD, then it is using this invention. Alternately, if the product has a mapping repository (such as a mapping file), then it is using this invention.

(7) Need by IBM: Is this invention now planned for use in IBM? If so how will it be used and for which products will it be used? For which IBM products does the invention have a potential use?

The invention is not planned yet for use in IBM.
It has a potential use in:

- The IBM DataJoiner.
- The IGS SRM/EPP (Server Resource Management/End-to-end Probe Platform) reporting tool.
- Applications that use diverse data sources.
- Any IGS solution for customers that are moving into the Web technology and need access to their legacy systems. This is a critical invention that can assist in those numerous solutions.
- Common Customer Information Architecture - new approach within IBM CRM which requires access to customer information which is currently distributed in many data files.

(8) Search (Complete if investigation of patent protection is recommended, and note if you have done any searching)

I've searched the IBM IP Network using the queries below:
diverse data sources
data integration
XML
data unification

(9) Prior Art: Please indicate any similar or related work that you are aware of, including URLS of applicable.

The closest found are:

- US5345586: Method and system for manipulation of distributed heterogeneous data in a data processing system
- US5884310: Distributed data integration method and system

There is no prior work on data integration using XML.